Available online at www.sciencedirect.com
ScienceDirect
Theoretical Computer Science 370 (2007) 34–47
Theoretical Computer Science
www.elsevier.com/locate/tcs

One Head Machines from a symbolic approach

A. Gajardo[a,*], J. Mazoyer[b]

[a] *Departamento de Ingeniería Matemática, Universidad de Concepción, Casilla 160-C, Concepción, Chile*
[b] *Laboratoire de l'informatique du parallélisme, École Normale Supérieure de Lyon, 46 Allée d'Italie, 69364 LYON Cedex 07, France*

Received 1 July 2005; received in revised form 27 June 2006; accepted 9 October 2006

Communicated by Bruno Durand

Abstract

We consider the Turing Machine as a dynamical system and we study a particular partition projection of it. In this way, we define a language (a subshift) associated to each machine. The classical definition of Turing Machines over a one-dimensional tape is generalized to allow for a tape in the form of a Cayley Graph. We study the complexity of the language of a machine in terms of realtime recognition by putting it in relation with the structure of its tape. In this way, we find a large set of realtime subshifts some of which are proved not to be deterministic in realtime. Sofic subshifts of this class correspond to machines that cannot make arbitrarily large tours. We prove that these machines always have an ultimately periodic behavior when starting with a periodic initial configuration, and this result is proved for any Cayley Graph.

Keywords: Turing Machine; Symbolic dynamics; Formal language; Discrete dynamical system

1. Introduction

Turing Machines (TMs) constitute the fundamental model for studying the power of algorithms. Their high simplicity makes their theoretical analysis possible. Many important results concerning the power and complexity of algorithms have been obtained through this model. Now we want to look at it from a dynamical point of view. This has a double interest. First, a TM can be considered as a particular case of a more general class of dynamical systems, those consisting of an automaton which moves over a network (a one-dimensional lattice in the case of TMs), interacting with the environment by reading and writing symbols on it. Some examples of this kind of system are the pebble automaton [3,5] and Langton's ant [4,6]. Secondly, it may allow us to compare the computational complexity and calculability of the TM with its dynamical properties.

TMs have already been studied from a dynamical approach in [9,2]. Since TMs were not conceived as a dynamical system, their original definition is not completely appropriate. So, a new definition is needed. There is not a unique way of doing this, and different ways will give the system different properties, as Kůrka notes [9].

We propose a new approach that takes advantage of a relevant feature of these systems: changes happen only on the automaton's position, while the rest of the space remains static. Thus, their entire evolution can be described by

* Corresponding author. Tel.: +56 412207042; fax: +56 412522055.
E-mail address: anahi@ing-mat.udec.cl (A. Gajardo).

0304-3975/$ - see front matter
doi:10.1016/j.tcs.2006.10.004

the linear sequence of states and symbols that the automaton reads, without ambiguity. Such a description creates a relationship between an n-dimensional dynamical system and a one dimensional symbolic system: the one given by the set of sequences that describe the automaton's behavior over the set of initial configurations.

After giving some elementary required notions of symbolic systems and Cayley Graphs (Sections 2 and 3) we define formally the proposed ideas and give some basic properties (Section 4). The results presented in this article are divided into two sections. In Section 5 we analyze the symbolic systems with reference to their complexity. More specifically, we study how the underlying graph can affect this complexity within the class of realtime subshifts. In Section 6, we characterize the property of the symbolic system of being sofic through some properties of the TM. We give some results which relate periodic points of both systems, and we show that if the symbolic system is sofic, then the TM has always an ultimately periodic behavior over periodic initial configurations.

2. Notions of symbolic dynamics

Words. An *alphabet* A is any nonempty finite set. Its elements are called *letters*. A *word* over A is a finite sequence of elements of A. Thus, the set of these words is $A^+ = \cup_{n\in\mathbb{N}} A^{n+1}$. Given two words u and v, the operation of concatenation is defined by uv. In this way $(A^+, \cdot)$ is a semigroup. By adding a neutral element called *the empty word* and denoted by 1, a monoid is obtained. It is denoted by $A^* = A^+ \cup \{1\}$.

If two words u and v satisfy the condition that there exist two words s, t such that $u = svt$, we say that v is a *factor* of u ($u \sqsubseteq v$). If, moreover, $s = 1$ ($t = 1$), then we say that v is a *left (right) factor* of u. The *length* of a word is denoted by $|u|$. Given a word u, we will in general assume that it is the sequence of letters $u_1u_2..u_{|u|}$. Its factors will be denoted by $u_{i..j} = u_iu_{i+1}..u_j$, with $i \le j$ and $i, j \in \{1, .., |u|\}$.

Topological dynamical systems. A topological dynamical system (TDS) is given by a pair (X, f), where X is a compact topological space and f is a continuous function from X to X. Given a TDS (X, f), a *subsystem* is a pair (Y, f) such that $Y \subset X$ is topologically closed and f-invariant, i.e., such that $f(Y) \subseteq Y$. Two TDSs (X, f) and (Y, g) are said to be *conjugated* if there exists a continuous and bijective function $\phi : X \to Y$ such that $\phi \circ f = g \circ \phi$. If the function ϕ is continuous and surjective, then Y is called a *factor* of X.

The objects of symbolic dynamics are the infinite sequences of letters, that is, the elements of $A^{\mathbb{N}}$. The operation of concatenation and the relation of factor can be extended in a natural way when infinite words are involved. Concatenation is defined only between finite words and infinite words in $A^{\mathbb{N}}$. Given an infinite sequence x (infinite word) and a finite word u, we say that u is a factor of x if there exists a finite word w and an infinite word y such that $x = wuy$.

In $A^{\mathbb{N}}$, a metric can be defined. Given two infinite words $x = (x_i)_{i\in\mathbb{N}}$ and $y = (y_i)_{i\in\mathbb{N}}$, if n is the smallest natural number such that $x_i \neq y_i$, the distance between x and y is defined by $d(x, y) = 2^{-n}$; if no such n exists because $x = y$, then $d(x, y) = 0$. This metric makes $A^{\mathbb{N}}$ compact. A particular operator, $\sigma : A^{\mathbb{N}} \to A^{\mathbb{N}}$, is defined by $\sigma((x_i)_{i\in\mathbb{N}}) = (x_{i+1})_{i\in\mathbb{N}}$. It is called the *shift* function. With the metric defined above, the shift function is a continuous function. Thus $(A^{\mathbb{N}}, \sigma)$ is a TDS, known as the *fullshift*. The subsystems of $(A^{\mathbb{N}}, \sigma)$ are called *subshifts*, or symbolic systems.

Languages and subshifts. A *language* is any subset of A^*. Given a subshift Y, we consider the language

$$L(Y) = \{w \in A^* : (\exists x \in Y)\ w \text{ is a factor of } x\}.$$

This is the *factors language* of Y. The factors language characterizes Y because $Y = \{x \in A^{\mathbb{N}} : (\forall u \sqsubseteq x)\ u \in L(Y)\}$. Another way to characterize a subshift is through a set of forbidden words. A language P is a *set of forbidden words* for Y if

$$Y = \{x \in A^{\mathbb{N}} : (\forall u \sqsubseteq x)\ u \notin P\}.$$

If Y has a finite set of forbidden words, Y is said to be a *shift of finite type (SFT)*.

Realtime hierarchy. A classification of subshifts has been introduced by Kůrka and Maass [10]. It defines what they call a *realtime* subshift. A realtime subshift is one whose factors language is recognized by a k-pushdown automaton that reads the input word at a rate of one symbol per time step, and it identifies a forbidden word exactly when reading its last symbol. These automata are less powerful than TMs.

A k-pushdown automaton is a structure $M = (A, \Omega, \Sigma_1, \Sigma_2, .., \Sigma_k, \lambda, o_0, F)$, where A is the input alphabet, Ω is the states set, Σ_i, for $i \in \{1, .., k\}$, is the alphabet of the i-th pushdown store, $\lambda \subset A \times (\Omega \times \Sigma_1^* \times \Sigma_2^* \times .. \times \Sigma_k^*)^2$ is a finite set representing the non-deterministic transition function, $o_0 \in \Omega$ is the initial state and $F \subset \Omega$ is the set of final states.

A labeled graph, G_M, is associated with M. Its set of vertices is the set $\Omega \times \Sigma_1^* \times \Sigma_2^* \times .. \times \Sigma_k^*$, and its labeled edges are

$$(e, s_1t_1, .., s_kt_k) \xrightarrow{a} (f, r_1t_1, .., r_kt_k)$$

where $(a, (e, s_1, .., s_k), (f, r_1, .., r_k)) \in \lambda$ is a rule of M and t_i are arbitrary words.

A k-pushdown automaton is said to be deterministic if, each time that we have two different rules $(a, (e, s_1, .., s_k), (f, r_1, .., r_k))$ and $(a, (e, s_1', .., s_k'), (f', r_1', .., r_k'))$ in λ such that for every i there exist words $t_i, t_i' \in A_i^*$ such that $s_it_i = s_i't_i'$, then we have also that $r_it_i = r_i't_i'$ and $f = f'$. This means that there is no ambiguity in the actions of the automaton. When an automaton is deterministic, we can define the set of rules λ as a partial function, by putting $\lambda(a, e, s_1, .., s_k) = (f, r_1, .., r_k)$ if and only if $(a, (e, s_1, .., s_k), (f, r_1, .., r_k)) \in \lambda$.

Given a k-pushdown automaton $M = (A, \Omega, \Sigma_1, \Sigma_2, .., \Sigma_k, \lambda, o_0, F)$, the language L_M accepted by M consists of all words u in A^* such that there exists a path in G_M with label u, starting on vertex $(o_0, 1, .., 1)$ and finishing on a vertex of the form $(e, t_1, .., t_k)$ with $e \in F$.

$\mathbb{R}(k)$ denotes the class of languages accepted by a deterministic automaton with k pushdown stores. $\mathbb{Q}(k)$ denotes the class of languages accepted by a non-deterministic automaton with k pushdown stores. Languages in $\mathbb{R} = \cup_k \mathbb{R}(k)$ are called *deterministic realtime languages*, and those in $\mathbb{Q} = \cup_k \mathbb{Q}(k)$ are called *non-deterministic realtime languages*. By abuse of notation, we continue to use the symbols $\mathbb{R}(k)$, $\mathbb{R}$, $\mathbb{Q}(k)$ and $\mathbb{Q}$ to denote the classes of subshifts whose factors languages belong to them.

As it is remarked in [10], if a subshift is recognized by a deterministic k-pushdown automaton, then it can be recognized by a deterministic k-pushdown automaton with the full set of final states: $F = \Omega$, therefore we will omit F from the definition of a deterministic automaton. A similar result is true for non-deterministic pushdown automata but, in this case, the number of stores may change.

The 0-pushdown automata are called *finite automata*. Deterministic and non-deterministic finite automata recognize the same class of languages: the class of *regular* languages, i.e., $\mathbb{R}(0) = \mathbb{Q}(0) = \text{Reg}$. The subshifts in $\mathbb{R}(0)$ are called *sofic*. It is straightforward that shifts of finite type are also sofic.

More details about *dynamical systems* and *subshifts* can be found in [1,8,12,11].

3. Notions of Cayley Graphs

A Cayley Graph is a directed graph that is defined over a group structure. It can be defined each time we have a finitely presented group. A *presentation* of a group G is a pair $\langle D|R\rangle$, where D is a set of elements of G that generates it (we assume that the inverses of the elements of D are all in D) and $R \subset D^*$ is a set such that G is isomorphic to the quotient of the free group D^* by the reflex, symmetric and transitive closure, $\equiv$, of the relation $\simeq$ defined by:

$$e \simeq f \Leftrightarrow (\exists r \in R)(\exists g, h \in D^*)\; e = grh \wedge f = gh.$$

Here, we use the multiplicative notation for the operation of the group, i.e., gh means g *operated with* h. Given a word $w \in D^*$, we will denote its equivalence class by $\overline{w}$ and we will consider it as an element of G.

The Cayley Graph of the presentation $\langle D|R\rangle$ is the pair (G, E) where $(g, h) \in E \Leftrightarrow (\exists d \in D)h = gd$. Given an element $g \in G$ we denote by $|g|$ the length of the shortest word $w \in D^*$ such that $\overline{w} = g$. In terms of the graph this equivalent to put $|g| = d(g, 1)$.

The same graph may correspond to different presentations of different groups. For example the 2-dimensional square grid is the Cayley Graph associated to two different groups: $\langle n, s, w, e|ns, sn, we, ew, nwse\rangle$ which represents $\mathbb{Z}^2$, and $\langle a, b, c, d|ac, ca, bd, db, a^4, b^4, adad, cbcb\rangle$ which is a non-commutative group ($cb \neq bc$). Fig. 1 shows the Cayley Graph of these groups, where the edges are labeled with the generators names in order to exhibit the difference between the groups.

We distinguish the Cayley Graphs of two important groups:

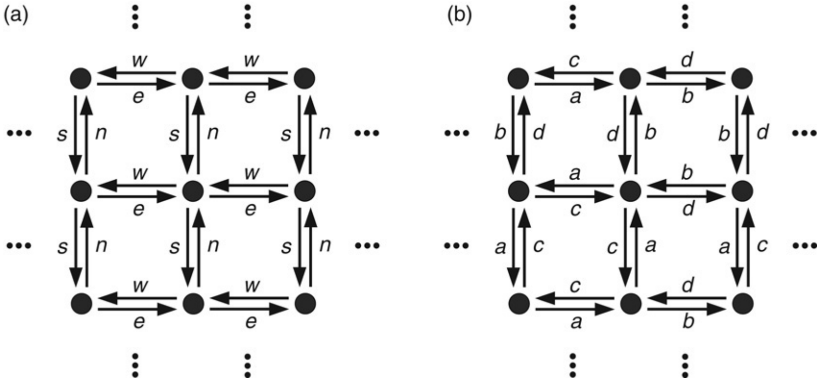

Fig. 1. These are the Cayley graphs of two different groups. (a) The 2-free abelian group: $\langle n, s, w, e | ns, sn, we, ew, nwse\rangle$. (b) The group presented by $\langle a, b, c, d | ac, ca, bd, db, a^4, b^4, adad, cbcb\rangle$.

The k-free group, which corresponds to a regular infinite tree of degree $2k$:

$$T = \langle \{a_i\}_{i=1}^{k}, \{a_i^{-1}\}_{i=1}^{k} | \{a_i a_i^{-1}\}_{i=1}^{k}\rangle.$$

The k-free abelian group, which corresponds to the k-dimensional cubic grid:

$$C = \langle \{a_i\}_{i=1}^{k}, \{a_i^{-1}\}_{i=1}^{k} | \{a_i a_i^{-1}\}_{i=1}^{k}, \{a_i a_j a_i^{-1} a_j^{-1}\}_{i,j=1}^{k}\rangle.$$

4. Definitions and basic results

We will study TMs whose tape is a Cayley Graph G. Thus, a symbol will be registered at each vertex of G. In other words, there will be a function $c : G \to S$ which we call *configuration*. The topology generated by the metric $d : S^G \times S^G \to \mathbb{R}$ defined by: $d(c, c') = 2^{-n}$, where $n = \min\{|g| : c(g) \neq c'(g)\}$, makes S^G compact.

Since we will study TMs from a dynamical systems point of view, we will consider a general definition in which we omit the start and halt states.

Definition 1. Given a finitely presented group, $G = \langle D|R\rangle$, a TM over it is a 5-tuple $(S, Q, D, G = \langle D|R\rangle, \delta = (\delta_Q, \delta_S, \delta_D))$, where:

S, Q and D are finite sets, respectively called the *symbol* set, the *state* set and the *directions* set;

G is a group with a finite presentation: $\langle D|R\rangle$; and

$\delta_i : S \times Q \to i$, with $i = S$, Q or D, are the local transition functions.

The elements of G are called *cells*. The state of the system is given by an assignment of symbols to each cell, $c : G \to S$, a position $g \in G$ and a state $q \in Q$, i.e., the phase space is $X = S^G \times G \times Q$.

The global transition function $T : X \to X$ is defined by $T((c, g, q)) = (c', g', q')$, where $q' = \delta_Q(c(g), q)$, $g' = g\delta_D(c(g), q)$, $c'(g) = \delta_S(c(g), q)$, and $c'(h) = c(h)$ for all $h \neq g$.

Taking into consideration the discrete topology for G, and the product topology for X, T trivially turns out to be continuous, thus defining a dynamical system (X, T). But X is not compact. There are several ways to define an equivalent system with a compact phase space. In [9], Kůrka proposes two different definitions of for TMs. They are not conjugated, and they have different dynamical properties.

Here we propose to study (X, T) through a symbolic system. We will call it *the trajectory subshift*, or *t-shift* for short. It consists in representing the state of the system through the sequence of states and symbols that the machine reads at each time step. This kind of representation is a particular case of a "partition subshift."

Definition 2. Given a TM $M = (S, Q, D, G, \delta)$ and its dynamical system (X, T), let $\pi : X \to S \times Q$ be defined by $\pi(c, g, q) = (c(g), q)$, and let $\psi : X \to S_T \subseteq (S \times Q)^{\mathbb{N}}$ be defined by $\psi(x) = (\pi(T^n(x)))_{n\in\mathbb{N}}$, where S_T is defined as the smallest set that well defines ψ, i.e., such that $\psi(X) = S_T$. The *t-shift* of (X, T) is S_T.

The language of S_T, $L(S_T)$ is called the *t-language* of T, and its words are called *t-words*.

The function ψ is not one-to-one, but the sequence $\psi(x)$ contains *all* the relevant information about the dynamics of x under T. The reason is that knowing the sequence of symbols and states of the machine is enough to recover its movement over G. Then we can deduce a kind of pre-image of ψ.

Definition 3. Given $w = \binom{\alpha_1..\alpha_n}{q_1..q_n}$ be a word in $(S \times Q)^*$. We define the sequence $I(w) \in G^{n+1}$ by recurrence:

$$I(w)_1 = 1,$$
$$I(w)_{j+1} = \overline{I(w)_j \delta_D(\alpha_j, q_j)}, \quad j \in \{1, .., n\};$$

the set:

$$V(w) = \{I(w)_j : 1 \leq j \leq n\};$$

and the partial configuration $c_w : V(w) \to S$ by

$$c_w(g) = \alpha_i \quad \text{where } i = \min\{j : I(w)_j = g\} \quad (\forall g \in V(w)).$$

Remark 1. (1) If we suppose that the machine starts at position $1 \in G$, the sequence $I(w)$ represents the itinerary of the machine, and $V(w)$ is the set of visited cells.
(2) Then we can deduce the initial state of these cells. It is given by the state that the machine finds the first time that it arrives at a cell. This justifies the definition of c_w.
(3) $I(uv)_{|uv|+1} = I(u)_{|u|+1} I(v)_{|v|+1}$.
(4) Any extension c' of c to G is a pre-image of w, in the sense that: if $x = (c', 1, q_1)$, we have that $(\psi(x))_{j=1}^n = w$, but this is true if and only if $w \in L(S_T)$; otherwise, w has no pre-image.
(5) These notions show a way to prove that ψ is continuous. It is also onto, and $\sigma \circ \psi = \psi \circ T$, then the system (S_T, σ) is a factor of (X, T).
(6) Definition 3 can be directly extended to infinite words.

This carries us to establish the following two conditions that characterize the words in $L(S_T)$. This property can be easily proved by induction.

Proposition 1. *If $w = \binom{\alpha_1..\alpha_n}{q_1..q_n} \in L(S_T)$, then for all $i \in \{1, .., n-1\}$:*

$$q_{i+1} = \delta_Q(\alpha_i, q_i) \qquad \textit{(state coherence)} \tag{1}$$

and for any pair $1 \leq i < j \leq n$, such that $I(w)_i = I(w)_j$ (say $= g$), and such that for every k between i and j, $I(w)_k \neq g$ one has that

$$\alpha_j = \delta_S(\alpha_i, q_i) \qquad \textit{(writing coherence).} \tag{2}$$

Moreover, these are sufficient conditions for w to belong to $L(S_T)$.

Eq. (1) expresses that the sequence of states must be coherent with the transition rule of the T M. Eq. (2) expresses that when the machine visits a cell for a second time, it must find the symbol that it wrote there when it visited it the first time. A set of forbidden words for S_T can be obtained from these two equations. The sets P_1 and P_2 are composed by the words that do not satisfy Eqs. (1) and (2), respectively.

$$P_1 = \left\{ \begin{pmatrix} \alpha\beta \\ qp \end{pmatrix} : p \neq \delta_Q(\alpha, q) \right\}$$
$$P_2 = \left\{ w = \begin{pmatrix} \alpha_1..\alpha_n \\ q_1..q_n \end{pmatrix} : I(w)_n = 1 \wedge 1 \notin \{I(w)_j\}_{j=2}^{n-1} \wedge \alpha_n \neq \delta_S(\alpha_1, q_1) \right\}.$$

A set of forbidden words for S_T is $P = P_1 \cup P_2$.

Remark 2. Given a deterministic k-pushdown automaton M that recognizes a t-shift S_T and given a vertex $\nu \in G_M$, we can assert that:

(1) If ν has an input edge labeled by (α, q), then all the exiting edges of ν have a label of the form $(\beta, \delta_Q(\alpha, q))$, with $\beta \in S$.
(2) Since every word in $L(S_T)$ defines a unique path in G_M starting at $(o_0, (e, .., e))$, Eq. (2) implies every path from $(o_0, (e, .., e))$ to a vertex v corresponds to a trajectory that has already visited its last cell or every path corresponds

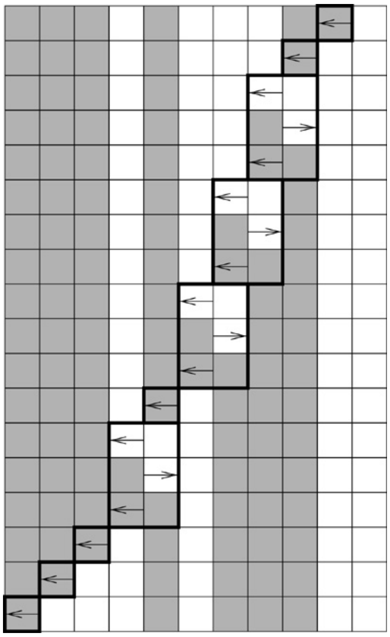

Fig. 2. A simulation of the 1D ant.

to a trajectory that has not visited its last cell before. In the first case, the out degree of v is one; in the second case it is $|S|$.

Example 1. Let us consider the TM representing Langton's ant in dimension one [7]. It has two states, $Q = \{Left, Right\}$; and two symbols, $S = \{White, Black\}$. It walks over $\mathbb{Z}$ with the following rule: if it is over a *Black* cell, it moves following its state, which it preserves. If it is over a *White* cell, it changes its state and follows its new state. It always switches the color of the cell. A space-time diagram is showed in Fig. 2.

We can prove that this machine can only make cycles of length two. Then the language of the 1-D ant is an SFT, because it has a finite set of forbidden words: $P = P_1 \cup P_2 = \left\{\binom{W}{L}\binom{*}{L}, \binom{W}{R}\binom{*}{R}, \binom{B}{L}\binom{*}{R}, \binom{B}{R}\binom{*}{L} : * \in \{W, B\}\right\} \cup \left\{\binom{W}{L}\binom{W}{R}\binom{W}{L}, \binom{B}{R}\binom{W}{R}\binom{B}{L}, \binom{W}{R}\binom{W}{L}\binom{W}{R}, \binom{B}{L}\binom{W}{L}\binom{B}{R}\right\}$. Fig. 3 shows the finite automaton that recognizes this language.

5. On the complexity of S_T

The class of complexity of S_T depends, of course, on T. We will see that the underlying graph of T is very relevant to this. In this section, we study the dependence between the underlying graph of T and the realtime hierarchy of S_T. We begin with a direct result, which states that $S_T \in \mathbb{R}(2)$ if T is a TM over $\mathbb{Z}$. Afterwards, we remark that, for the most common graphs (in particular the k-free group and the k-free abelian group) $S_T \in \mathbb{Q}$. The main part of this section is devoted to proving the existence of machines such that $S_T \in \mathbb{Q} \setminus \mathbb{R}$.

The proof of the following proposition is simple, and consists in observing that one can infer and register the content of the tape of the machine into two pushdown stores when receiving a t-word.

Proposition 2. *The language of a machine over $\mathbb{Z}$ is recognizable, in realtime, by a deterministic 2-pushdown automation.*

We can generalize this result by asserting that the t-language of a TM over a graph G can always be recognized in realtime by another TM over the same graph.

In general, the algorithmic complexity of $L(S_T)$ is related to the complexity of the word problem[1] of the underlying graph. In fact, the following algorithm decides whether or not a word w is in $L(S_T)$.

[1] The word problem is the problem of determining whether a word in D^* is equivalent to 1 in G or not.

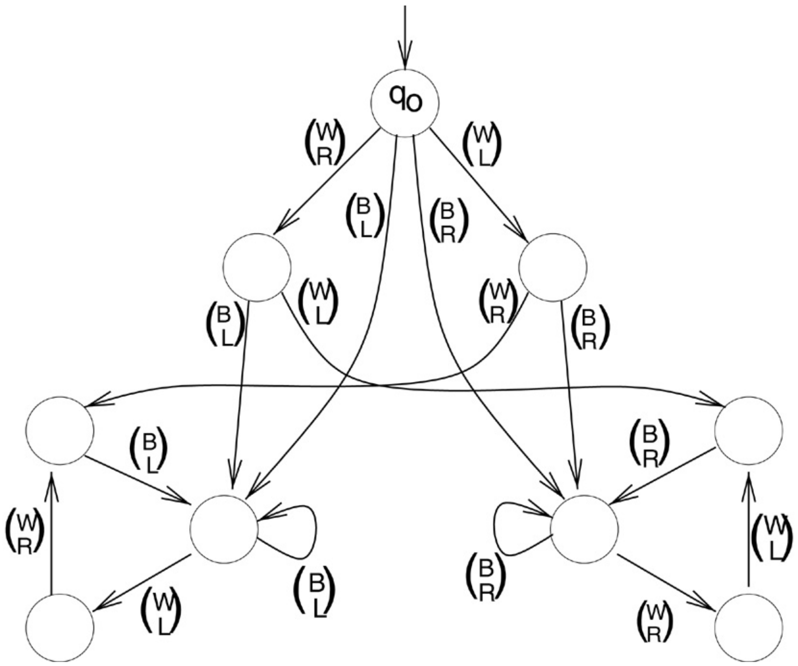

Fig. 3. A deterministic automation that recognizes the language associated to the 1D ant.

- Verify condition (1) for every factor of w of length two
- for each factor u of w, verify $I(u)_{|u|} = 1$ in G
  · if yes, verify condition (2).

Then $L(S_T)$ can be solved in $o(|w|^2 t(w))$ iterations, where $t(w)$ is the complexity of the word problem for the given underlying graph. The algorithm above is not realtime, and therefore this observation does not give information about the realtime complexity of $L(S_T)$. Nevertheless, the space needed in this algorithm is $o(s(w))$, where $s(w)$ is the space complexity of the word problem, i.e., $L(S_T)$ has the same space complexity of the word problem of G. In the case of the k-free abelian group and the k-free group, the word problem is linear space bounded. Thus $L(S_T)$ is also linear space bounded, which implies that S_T is realtime for these groups. We can state the following proposition.

Proposition 3. *The language of a machine over the k-free abelian group or the k-free group is realtime.*

The remainder of this section provides a detailed exposition of a machine in the 2-free abelian group ($\mathbb{Z}^2$) whose t-language is not deterministic realtime. A similar construction is provided for a machine over the k-free group. The proofs are based on a useful concept about languages that we recall in the following.

Let L be a language over the alphabet A. For each $n \in \mathbb{N}$, we define the relation R_n in L by:

$$uR_n v \Leftrightarrow (\forall w \in A^n)[uw \in L \Leftrightarrow vw \in L]$$

R_n is an equivalence relation. The cardinality of the quotient set $|L/R_n|$ is related to the amount of memory that the automaton may need to look up in order to recognize the remaining n symbols of any word. The following well-known properties hold.

Property 1. *If $M = (A, \Omega, \lambda, o_0)$ is a deterministic finite automaton that recognizes L, then $(\forall n \in \mathbb{N})\ |L/R_n| \leq |\Omega|$.*

Property 2. *If L is recognized in real time by a deterministic automaton $M = (A, \Omega, \Sigma_1, .., \Sigma_k, \lambda, o_0)$, then for all $n \in \mathbb{N}$.*

$$|L/R_n| \leq |\Omega|(|\Sigma_1||\Sigma_2|..|\Sigma_k|)^{mn}$$

where $m = \max\{|s_i| - |r_i| \ : i \in \{1, .., k\} \wedge (a, (e, s_1, .., s_k), (f, r_1, .., r_k)) \in \lambda\}$.

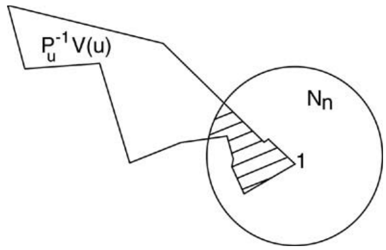

Fig. 4. The TM can go only on the shadowed part of $p_u^{-1}V(u)$ if it starts at 1.

The next property provides a bound on $|L(S_T)/R_n|$, which depends on the growing rate of the underlying Cayley Graph of T. The given upper bound is over exponential. Then, if it is attained, it is a proof that the language is not in $\mathbb{R}$.

Proposition 4. *The language of a machine, $M = (S, Q, D, G, \delta)$ with dynamical system (X, T), satisfies the condition that:*

$$|L(S_T)/R_n| \leq |Q|(|S|+1)^{|N_n|}$$

where $N_n = \{\overline{w} : |w| \leq n, w \in D^*\}$ is the neighborhood of radius n of the origin in the Cayley Graph of G.

Proof. The idea is to associate a different element of $Q \times (S \cup \{*\})^{N_n}$ with each equivalence class of $L(S_T)/R_n$, where $*$ is a fixed symbol which is not in S.

Given $u = \binom{\alpha_1(u)..\alpha_{|u|}(u)}{q_1(u)..q_{|u|}(u)} \in L(S_T)$, we consider $I(u)$, $V(u)$ and $c_u : V(u) \to S$.

Let us define $p_u = I(u)_{|u|+1}$ and $d_u : p_u^{-1}V(u) \to S$ by $d_u(g) = c_u(p_u g)$. The configuration d_u is the translation of c_u by p_u^{-1}. Thus, if the TM T starts on p_u^{-1} over any extension of d_u, it produces u and it stops at cell 1 in $|u|$ steps. In other words, if d'_u is an extension of d_u to G, then $\psi(d'_u, p_u^{-1}, q_1(u))_{1,..,|u|} = u$. Let us also define the resulting configuration b_u by:

$$(b_u, 1, \delta_Q(\alpha_{|u|}(u), q_{|u|}(u))) = T^{|u|}(d'_u, p_u^{-1}, q_1(u)).$$

Let us remark now that given a word $w \in (S \times Q)^n$, we have that $uw \in L(S_T)$ if and only if the following three conditions hold:

$w \in L(S_T)$;

$c_w|_{p_u^{-1}V(u)\cap V(w)} = b_u|_{p_u^{-1}V(u)\cap V(w)}$; and

$\delta_Q(\alpha_{|u|}(u), q_{|u|}(u)) = q_1(w)$.

That is, w is itself a t-word, and the configuration that produces w coincides with b_u in the intersection of their domains. Finally, it is important to respect the state coherence. But $V(w) \subset N_n$ (see Fig. 4). Thus, in order for u and v to be related it is enough to have that:

$p_u^{-1}V(u) \cap N_n = p_v^{-1}V(v) \cap N_n$;

$b_u|_{p_u^{-1}V(u)\cap N_n} = b_v|_{p_u^{-1}V(u)\cap N_n}$; and

$\delta_Q(\alpha_{|u|}(u), q_{|u|}(u)) = \delta_Q(\alpha_{|v|}(u), q_{|v|}(v))$.

Then, with each equivalence class $[u]$, we can associate the state $q = \delta_Q(\alpha_{|u|}(u), q_{|u|}(u))$, and the configuration $s_u : N_n \to S \cup \{*\}$ defined by $s_u(g) = b_u(g)$ if $g \in p_u^{-1}V(u)$, and $s_u(g) = *$ otherwise. If $[u]$ and $[v]$ are different classes, then one of the conditions above will fail, and either a different q or a different s will be assigned to each class. $\square$

The following shows that the former bound is attained, at least on a square grid.

Proposition 5. *There exists a TM over the 2-dimensional square grid T such that:*

$$|L(S_T)/R_n| \geq 2^{\frac{n^2-n}{2}}.$$

$$
\begin{array}{ccccc}
\multicolumn{5}{c}{\text{The first component}} \\
\rightarrow & \downarrow & \rightarrow & \downarrow & \uparrow \\
 & \downarrow & \uparrow & \downarrow & \uparrow \\
 & \rightarrow & \uparrow & \downarrow & \uparrow \\
 & & & \downarrow & \uparrow \\
 & & & \rightarrow & \uparrow
\end{array}
\qquad
\begin{array}{ccccc}
\multicolumn{5}{c}{\text{The second component}} \\
\downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
 & \downarrow & \downarrow & \downarrow & \downarrow \\
 & \downarrow & \downarrow & \downarrow & \downarrow \\
 & & & \downarrow & \downarrow \\
 & & & \downarrow & \downarrow
\end{array}
\qquad
\begin{array}{ccccc}
\multicolumn{5}{c}{\text{The last component}} \\
b_{11} & b_{12} & b_{13} & b_{14} & b_{15} \\
 & b_{22} & b_{23} & b_{24} & b_{25} \\
 & 1 & b_{33} & b_{34} & b_{35} \\
 & & & b_{44} & b_{45} \\
 & & & 1 & b_{55}
\end{array}
$$

Fig. 5. A sketch of the tree components of the initial configuration that defines u and v.

Proof. The idea of the proof is to define a particular TM, and $2^{\frac{n^2-n}{2}}$ words in its t-language, each one in a different R_n class. We define the words through the initial configuration that generates them. We choose a TM which does not write, then we do not define δ_S.

$$
\begin{aligned}
&M = (S, Q, D, G, \delta) \\
&S = \{\uparrow, \downarrow, \leftarrow, \rightarrow\}^2 \times \{0, 1, 2\} \\
&Q = \{A, B\} \\
&D = \{\uparrow, \downarrow, \leftarrow, \rightarrow\} \\
&G = \langle \uparrow, \downarrow, \leftarrow, \rightarrow \mid \uparrow\downarrow, \downarrow\uparrow, \leftarrow\rightarrow, \rightarrow\leftarrow, \uparrow\leftarrow\downarrow\rightarrow \rangle \ \ (=\mathbb{Z}^2) \\
&(\delta_Q(q, (f, r, a)), \delta_D(q, (f, r, a))) = \begin{cases} (B, f) & \text{if } q = B \wedge a \in \{0, 1\} \\ (A, f) & \text{if } q = B \wedge a = 2 \\ (A, r) & \text{if } q = A. \end{cases}
\end{aligned}
$$

If the machine is on state B, it follows the direction indicated by the first component of the symbol. It passes from state B to A if it reads a '2' in the third component of the symbol. When in state A, it follows the direction indicated by the second component of the symbol.

Given $(b_{ij})_{i=1,j=1}^{n-1} \in \{0, 1\}$, we define an initial configuration such that the first component of the symbol makes the TM follow a zigzag trajectory over a triangle. The second component of the symbol makes the TM move straight from north to south. And the last component of the symbol is given by (b_{ij}). Fig. 5 shows an example for $n = 6$.

If the machine starts on cell $(1, 1)$ on state B, it follows a zigzag trajectory which covers the triangle and exits by the north of cell $(1, n-1)$. In that way we define $2^{\frac{n^2-n}{2}}$ different words in $L(S_T)$. Now we need to show that they are not R_n related. Let u and v be generated in this way. In order to prove that u is not related with v, it is enough to exhibit a word w of length n such that uw is in $L(S_T)$ and vw is not. Let us suppose that the difference between u and v is on the column j of the initial configuration.

We can define the configuration outside the triangle in such a way that after the machine exits it, it borders it to the north side, up to column j, where it finds a symbol with a number 2 in its last component, and makes the machine follow the second component of the symbols and enter the triangle. The machine will find the symbol that makes u different from v, and it will do this before n steps after exiting the triangle by the first time. This proves that u and v are not R_n related and concludes the proof. □

The previous proof can be generalized to any k-free abelian group, showing that the class of t-shift corresponding to machines of dimension k is strictly contained in the class corresponding to dimension $k + 1$.

On the other hand, it is not possible to generalize this proof to any Cayley Graph in order to obtain a lower bound of order $o(2^{N_n})$ of $|L(S_T)/R_n|$. Two difficulties appear. First, it is necessary to define a TM that visits exhaustively N_n. This cannot be done in a general way. Second, the TM needs to be able to reach $o(|N_n|)$ points from a given neighborhood in less than n steps (with different trajectories, but with a big part of the configuration fixed). In the case of the square grid this was possible, since a quarter of the neighborhood can be covered by a family of n geodesics which are disjoint over an important part. But this is not the general case. In the case of trees, k^n different geodesics are needed to cover N_n, and they are not disjoint.

The following proposition is a generalization for the case of trees that overcomes the first problem. Since the second problem cannot be solved, we give a weaker bound for $|L(S_T)/R_n|$ in this case. Nevertheless, the proposition shows a machine whose t-shift is not deterministic realtime. It will be proved in much the same way as Proposition 5.

Proposition 6. *There exists a TM over the k-free group, T, such that*

$$|L(S_T)/R_{2n^2-n}| \geq 2^{(2k-1)^{n-1}}.$$

Proof. We only give the idea of the proof. As before, the TM's alphabet will have three coordinates. One has the function of guiding the machine to make it visit the set $N_n^d = \{\overline{dw} : w \in D^* \wedge w_1 \neq d^{-1} \wedge |w| = n-1\}$, with $d \in D$. This set can be visualized as a branch of radius n of the k-free group. The second is used by the machine in the second stage of the procedure. The last one contains information.

In a first stage, the machine visits all the cells of N_n^d. Each cell contains a bit of information, defining in this way $2^{(2k-1)^{n-1}}$ different words in $L(S_T)$. The TM stops at the origin; it exits the branch and passes to a new state, in which it follows instructions written in the second coordinate of the tape symbols. The TM will need to write on the tape to follow this instructions.

Another branch contains the information describing a trajectory of length n in the first visited branch. The TM needs to oscillate between these two branches in order to transcribe this information and to follow the indicated trajectory. At the end, all the cells in the given trajectory will be visited. This operation takes $2n^2 - n$ iterations.

Since any trajectory of length $n-1$ may be described in the second branch, the $2^{(2k-1)^{n-1}}$ words belong to different R_{2n^2-n} classes. $\square$

These propositions prove that in general the language of S_T cannot be recognized with a deterministic j-pushdown automaton in realtime, i.e., $S_T \notin \mathbb{R}(j)$ for every j. We conclude with the following corollary.

Corollary 1. *For any $k \geq 2$, there exists a machine over the k-free group and a machine over the k-free abelian group, such that their t-shifts are in $\mathbb{Q} \setminus \mathbb{R}$.*

In the light of Propositions 4 and 5, we remark that if the t-shift, S_T, of a TM over the square grid is in $\mathbb{R}(k)$, for some k, then the TM needs to have restrictions on its movement ability. In some sense, the machine cannot come back in a rapid way to any visited cell if S_T is in $\mathbb{R}$, and it cannot come back too far if S_T is in $\mathbb{R}(0)$. We analyze these observations deeply in next section.

6. Dynamical properties of S_T and T

In this section, we explore periodic points of both systems and their respective images. Obviously, a periodic point of T is also a periodic point of S_T. The converse is not true in general, but we will show that some properties of pre-images of periodic points in S_T can be established.

Soficity can be completely characterized on $\mathbb{Z}$ through dynamical properties of T. Over an arbitrary graph, only systems of the finite type can be well characterized. The main result of this section deals with the dynamical properties of machines whose t-language is sofic. More precisely, we prove that if S_T is sofic, then T always has an eventually regular behavior when it starts over a periodic configuration. In this section, we always consider that (X, T) is the dynamical system of the machine $(S, Q, D, G = \langle D|R\rangle, \delta_S, \delta_Q, \delta_D)$. Let us recall that $X = (S^G, G, Q)$.

Proposition 7. *If (c, g, q) is a periodic point of (X, T), then $\psi(c, g, q)$ is a topologically isolated point in S_T.*

Proof. Since (c, g, q) is a periodic point, there exists $t \in \mathbb{N} \setminus \{0\}$ such that $T^t(c, g, q) = (c, g, q)$. Let us define $x = \binom{\alpha_i(x)}{q_i(x)}_{i\in\mathbb{N}} = \psi(c, g, q)$, and let $n > t$ be a natural number. We assert that the cylinder $[x_{1..n}] \cap S_T$ has only one point: x. Let us suppose that there exists $y \in [x_{1..n}] \cap S_T$, with $y = \binom{\alpha_i(y)}{q_i(y)}_{i\in\mathbb{N}} \neq x$. Let $i \in \mathbb{N}$ be the smallest index such that $y_i \neq x_i$; of course $i > n$. It is necessary that $I(y)_i = I(x)_i$, because the direction that the machine took at iteration $i-1$ was the same for both words. Also $q_i(x) = q_i(y) = \delta_Q(x_{i-1})$, this implies that $\alpha_i(x) \neq \alpha_i(y)$. But $V(x) = V(x_{1..n})$, and therefore $I(y)_i \in V(x_{1..n}) = V(y_{1..n})$. This means that $I(y)_i$ has already been visited by the machine before step i. Then, the machine finds the same symbol in $I(y)_i$ in both trajectories y and x, which contradicts the assumption that $y_i \neq x_i$. $\square$

This property has a direct consequence over the transitivity of S_T. Let us recall that a symbolic system is *transitive* if for every pair of words u and v of the language, there exists a word w such that uwv is also in the language.

Corollary 2. *If (X, T) has a periodic point, then S_T is not transitive.*

The converse is not true as the subshift shown in Example 1 shows. It is composed by two disjoint and conjugated transitive subsystems.

Let us establish now the properties of the pre-images of the periodic points of S_T. Some of them are periodic too (in some sense), but let us define first what periodicity means in S^G.

Definition 4. Given $g \in G$, we define the *g-shift function* $\sigma_g : S^G \to S^G$ by $\sigma_g(c)(p) = c(gp)$. We will consider three levels of periodicity:

(1) Given $g \in G$, a function $c : G \to S$ is *g-periodic* if $\sigma_g(c) = c$.
(2) A configuration c is said to be *periodic* if it is g-periodic for some $g \in G$ such that $(\forall k \in \mathbb{N})\ g^{k+1} \neq 1$.
(3) A configuration c is said to be *strongly periodic* if the set $\{\sigma_g(c) : g \in G\}$ is finite.

We will say that a configuration c is *(strongly) periodic except for a finite number of cells*, if there exists a (strongly) periodic configuration a that differs from c only in a finite number of cells.

Proposition 8. *If $x \in S_T$ is periodic, then there exists an initial configuration c that is periodic except for a finite number of cells, which produces x for some initial state and position.*

Proof. Let $x = \binom{\alpha_i}{q_i}_{i \in \mathbb{N}} \in S_T$ be a word of period t, i.e., $x = uu..$, with $|u| = t$. Two cases appear.

Case 1. $V(x)$ is finite. In this case, the machine visits a finite number of cells, and the result is trivial.

Case 2. $V(x)$ is infinite. Let us define $g = I(x)_t$. This implies, by Remark 1, that $I(x)_{kt+i} = g^k I(x)_i$, for every $k \in \mathbb{N}$, $i \in \{0, .., t\}$. Then $V(x) = \cup_{k \in \mathbb{N}} g^k V(u)$. This means that $V(x)$ is the union of translations of $V(u)$ by g^k. Of course, $g^k \neq 1$, for every $k \in \mathbb{N} \setminus \{0\}$, otherwise $V(x)$ would not be infinite.

Assertion. The configuration c_x is eventually periodic in the direction g, i.e.,

$$(\exists \tau \in \mathbb{N})(\forall f \in V(u))(\forall k \geq \tau)\ c_x(g^k f) = c_x(g^\tau f).$$

Proof of the assertion. Let us define the configuration $b_x : V(x) \to S$ resulting from making the machine work over c_x during t iterations, that is, $(b_x, g, q_0) = T^t(c_x, 1, q_0)$. Then, $\psi(b_x, g, q_0) = x$. Two configurations that produce the same word (x) must coincide over the cells that the machine visits, modulo a translation. In this case, this translation is g. Then b_x satisfies $b_x(gh) = c_x(h)$ for every $h \in V(x)$. Moreover, within the first t iterations, the machine modifies only the cells in $V(u)$, and therefore $b_x|_{V(x)\setminus V(u)} = c_x|_{V(x)\setminus V(u)}$. Then, if we define

$$\tau = 1 + \max\{k | g^k V(u) \cap V(u) \neq \phi\}.$$

It is not difficult to see, by induction on k, that if $k \geq \tau$ then $c_x(g^k f) = c_x(g^\tau f)$ for every $f \in V(u)$, which is the assertion.

It remains to prove the existence of τ. Let us suppose that it does not exist, then there exists a strictly monotone sequence $(k_i)_{i \in \mathbb{N}}$ of natural numbers, and a sequence $(f_i)_{i \in \mathbb{N}}$ of cells in $V(u)$ such that $g^{k_i} f_i \in V(u)$. The sequence $(f_i)_{i \in \mathbb{N}}$ remains inside the finite set $V(u)$, and thus there exists a cell f and an infinite subsequence $(f_{i_j})_{j \in \mathbb{N}}$ such that $f_{i_j} = f$, for every $j \in \mathbb{N}$.

Since the sequence $(g^{i_j} f)_{j \in \mathbb{N}}$ remains inside the finite set $V(u)$, there exist two integers $k < j$ such that $g^{i_j} f = g^{i_k} f$. This implies that $g^{i_j - i_k} = 1$, which is impossible.

The assertion implies that c_x is g-periodic, except for the finite set $\bigcup_{k=0}^{\tau} g^k V(u)$. But c_x is only defined in $V(x)$. Now, we need to define a g-periodic configuration in G, differing from c_x only over a finite set. Let us fix $s \in S$, and let us define the g-periodic configuration a.

$$a(h) = \begin{cases} s & \text{if } h \notin \{g^k V(u)\}_{k \in \mathbb{Z}} \\ c_x(g^\tau h) & \text{if } h \in \{g^k V(u)\}_{k \in \mathbb{N}} \\ c_x(g^{i+\tau} h) & \text{if } h \in g^{-i} V(u) \setminus \{g^{k-i+1} V(u)\}_{k \in \mathbb{N}} \text{ for some } i \geq 1. \end{cases}$$

Let us prove that a is g-periodic. Let $h \in G$. If h is not in $\{g^k V(u)\}_{k \in \mathbb{Z}}$, then neither is gh. Therefore $a(gh) = s = a(h)$.

If h is in $\{g^k V(u)\}_{k \in \mathbb{N}}$, then gh is too, and thus $a(gh) = c_x(g^{\tau+1} h)$, which is equal to $c_x(g^\tau h) = a(h)$ from the assertion.

If h is in $g^{-i}V(u) \setminus \{g^{k-i+1}V(u)\}_{k\in\mathbb{N}}$ for some $i \geq 2$, then $gh \in g^{-(i-1)}V(u) \setminus \{g^{k-(i-1)+1}V(u)\}_{k\in\mathbb{N}}$. Thus $a(gh) = c_x(g^{i-1+\tau}gh) = a(h)$. If $i = 1$, then $gh \in V(u)$ and $a(gh) = c_x(g^{\tau}h) = c_x(g^{\tau+1}h) = a(h)$.

The configuration a can differ from c_x only in a finite number of cells. Extending c_x to G by the values of a defines a periodic configuration, except for a finite number of cells which constitute a pre-image of x. □

Now let us study sofic systems and shifts of a finite type. It is clear that if the time between two visits of the machine to the same cell is bounded, then the system S_T is of finite type. Let us prove the converse.

Proposition 9. *S_T is of finite type if and only if the time between two consecutive visits of the machine to the same cell is bounded.*

Proof. Let us suppose that S_T is of finite type. Let $P \subset (S \times Q)^*$ be a finite set of forbidden words for it. Let n be the length of its longest word.

Let $w = w_1..w_m$ be a word corresponding to a closed trajectory of T with $m \geq n + 2$, and such that it visits the first cell exactly two times: at iterations 1 and m. Let us suppose that $w_i = \binom{\alpha_i}{q_i}$ for every i. The word $u = w_2..w_{m-1}$ belongs to the language of S_T.

Let us analyse the trajectory that produces u, supposing that the machine starts on cell $I(w)_2$. Since T will move to cell 1 at iteration $m - 1$ and this cell has not been visited within u, the word $u\binom{\alpha}{q_m}$ must belong to the language of S_T for every $\alpha \in S$.

The word $w_1w_2..w_{m-1}$ belongs to the language of S_T too. Since there are not words of length $m - 1$ in P, the word $w_1w_2..w_{m-1}\binom{\alpha}{q_m}$ must belong to the language of S_T for any $\alpha \in S$.

But this is not possible since the symbol at cell 1 is $\delta_S(\alpha_1, q - 1)$ and no other one should be accepted. □

If a system is of finite type, it is also sofic, but the converse is not true in general. Nevertheless, if the system is the trajectory shift of some 1-dimensional TM, then to be sofic is equivalent to being of finite type.

Proposition 10. *If T is a TM over the group $G = \mathbb{Z}$, then S_T is sofic if and only if the time between two consecutive visits of the machine to the same cell is bounded.*

Proof. Let us suppose that S_T is sofic. Let $M = (S \times Q, \Omega, \lambda, o_0)$ be a finite automaton that recognizes S_T.

The cells of $\mathbb{Z}$ can be represented by numbers; the origin will be represented by the number 0 rather than 1, as was the case for other groups. The set of generators for $\mathbb{Z}$ will be $D = \{-1, 1\}$, so that the machine can move only one unit to the left or to the right respectively.

If the number of cells that a closed trajectory visits is bounded, then the length of a closed trajectory is also bounded. We will prove that if S_T is sofic, then this first number ($|V(w)|$ when w is closed and visits 0 only twice) is bounded.

Let $w = w_1..w_m \in L(S_T)$ be a word such that $I(w)_1 = I(w)_m = 0$ and $(\forall i \in \{2, .., m - 1\})\ I(w)_i \neq 0$. Let us suppose that $|V(w)| > |\Omega|$ and $I(w)_2 = 1$ (i.e., all the machine's trajectory is at the right of 0). Let $i = \operatorname{argmax}\{I(w)_j : 1 \leq j \leq m\}$. This implies that for every $k > i$ (and $k \leq m$) the cell $I(w)_k$ has already been visited before iteration i. Naturally, $m - i \geq |V(w)|$.

Since M is a deterministic automaton, the word w corresponds to a unique sequence of vertices of Ω which starts on $o_0 : o_0o_1..o_m$. Since $m - i > |\Omega|$, there exist two indices r and t such that $i \leq r < t \leq m$ and $o_r = o_t$. The outdegree of the vertices o_j is 1 for every $j \geq i$. Then there is a unique possible continuation of $o_0..o_m$, and it repeats the sequence $o_r..o_t$ periodically. This implies that w has a unique possible continuation too, and moreover, that all the cells that the machine will visit after r are in $\{0, .., I(w)_i\}$.

If $I(w)_r = I(w)_t$, then the TM has a bounded trajectory which does not include the cell 0, which is not possible.

If $I(w)_r \neq I(w)_t$, then the machine will visit an infinite number of cells, which is impossible. Then w cannot exists. □

The same proof does not hold for the square grid. In fact, it is possible to define a TM that cannot revisit more than one cell in all its trajectories and that can make cycles of arbitrary length. Nevertheless, it is difficult to imagine a set of cycles of arbitrary length that can be recognized by a finite automaton. The result may be true for general Cayley Graphs.

The following results discuss the dynamics of (X, T) when S_T is sofic.

Theorem 1. *If $G = \mathbb{Z}$ and S_T is sofic, then $x = \psi(c, g, q_0)$ is eventually periodic whenever c is periodic except for a finite number of cells.*

Proof. Let $M = (S \times Q, \Omega, \lambda, o_0)$ be a deterministic finite automaton that recognizes S_T. Let $c : \mathbb{Z} \to S$ be a l-periodic configuration except for a finite number of cells, i.e., $c = ..dddd.bdddd..$, with $|d| = l$. We can suppose without loss of generality that l divides $|b|$ and g. Let $x = \binom{\alpha_i}{q_i}_{i\in\mathbb{N}} = \psi(c, g, q_0)$.

Let us consider the automaton $\overline{M} = (S \times Q, \Omega \times \{0, .., l-1\}, \overline{\lambda}, (o_0, 0))$, where a rule $\left(\binom{\beta}{q}, (\mu, i), (\nu, j)\right) \in \overline{\lambda}$ if and only if $\left(\binom{\beta}{q}, \mu, \nu\right) \in \lambda$ and $j = i + \delta_D(\beta, q) \bmod l$.

$\overline{M}$ recognizes the same language as M. Its set of vertices $\Omega \times \{0, .., l-1\}$ registers the position of the TM over $\mathbb{Z}$ modulo l. In fact, let us suppose that y is the label of the sequence of vertices $((o_k, p_k))_{k\in\mathbb{N}}$. If $k = 0$, the position of the machine is a multiple of l and $p_0 = 0$. If $I(y)_i = p_{i-1} \bmod l$ for some i and $y_i = \binom{\beta}{q}$, then $I(y)_{i+1} = p_{i-1} + \delta_D(\beta, q) \bmod l$, which is equal to p_i by definition of $\overline{\lambda}$. Observe that the degree of o in M is equal to the degree of (o, p) in $\overline{M}$.

Now let us study x. Two possibilities appear: $V(x)$ can be finite or not. If it is finite, $(T^n(c, g, q_0))_{n\in\mathbb{N}}$ is eventually periodic and so it is x. Let us analyze the case where $V(x)$ is infinite. From Proposition 10 we know that the number of cells visited during a closed finite trajectory is bounded. Let N be an upper bound of this quantity. This means that if the machine is at a distance $N+1$ from a cell that it has already visited, the machine will not visit this cell anymore.

Since $V(x)$ is infinite, we can choose an iteration k such that $|I(x)_k| > |b| + N$. This assures that the cell $I(x)_i$ has already been visited at iteration i, or its state is given by $d_{I(x)_i \bmod l}$ (when $i > k$).

Let $(o_0, 0), (o_1, p_1), (o_2, p_2), ..$ be the sequence of vertices in $\Omega \times \{0, .., l-1\}$ whose label is x, i.e., $(x_i, (o_{i-1}, p_{i-1}), (o_i, p_i)) \in \overline{\lambda}$. There are two kinds of vertices. Either $\deg((o_{i-1}, p_{i-1})) = 1$ or $\deg((o_{i-1}, p_{i-1})) > 1$ in the graph of $\overline{M}$. If $deg((o_{i-1}, p_{i-1})) > 1$, $I(x)_i$ is being visited for the first time at iteration i. If in addition $i > k$, then $c(I(x)_i) = d_{I(x)_i \bmod l} = d_{p_{i-1}}$, and then $x_i = \binom{d_{p_{i-1}}}{q_i}$. By Remark 2, the state q_i is determined by the vertex o_{i-1}. This means that (o_i, p_i) is uniquely determined by (o_{i-1}, p_{i-1}), for every $i > k$.

Now, since $\Omega \times \{0, .., l-1\}$ is a finite set, there will be two iterations $i, j > k$ such that $(o_i, p_i) = (o_j, p_j)$, and then $(o_{i+r}, p_{i+r}) = (o_{j+r}, p_{j+r})$ for every $r > 0$. That is to say, the sequence of vertices is eventually periodic, which implies that the sequence of labels x is eventually periodic too, concluding the proof. □

As a corollary we obtain the following intuitive result, which relates the complexity of the TM with the complexity of its language.

Corollary 3. *If S_T is sofic, then T cannot be universal.*

Theorem 1 can be generalized to an arbitrary Cayley graph with some additional hypotheses.

Theorem 2. *If S_T is sofic, then $x = \psi(c, f, q_0)$ is eventually periodic whenever c is strongly periodic.*

Proof. The idea of the proof is analogous to that of Theorem 1. We define a new automaton $\overline{M}$ with a set of states of the form $\Omega \times H$, where H is a finite set, which allows us to describe completely the initial configuration. The definition of H is quite delicate because the underlying group may even be non-abelian.

Let $c : G \to S$ be a strongly periodic configuration. Then $\{\sigma_g(c) | g \in G\}$ is finite. Then we can define H as the quotient set of G by the following equivalence relation:

$$g \sim h \Leftrightarrow \sigma_g(c) = \sigma_h(c).$$

Then H is a finite set. In the general case, H is not a group. Nevertheless, an action can be defined on it: given $a, h \in G$, we define $[h] \star a = [ha]$. It is well defined, in fact, if $g \sim h$ then $\sigma_{ha}(c)(x) = c(hax) = \sigma_h(ax) = \sigma_g(ax) = c(gax) = \sigma_{ga}(c)(x)$.

Also c is well defined on H by putting $c([g]) = c(g)$. In fact, if $g \sim h$, then $c(g) = \sigma_g(c)(1) = \sigma_h(c)(1) = c(h)$. In this way, the configuration c is described by its values over H, which are finite in number.

Now let $M = (S \times Q, \Omega, \lambda, o_0)$ be the automaton that recognizes S_T. We define $\overline{M} = (S\times, \Omega \times H, \overline{\lambda}, (o_0, [1]))$, where $\overline{\lambda}$ is defined by:

$$\left(\binom{\alpha}{q}, (\mu, j), (\nu, i)\right) \in \overline{\lambda} \Leftrightarrow \left(\binom{\alpha}{q}, \mu, \nu\right) \in \lambda \quad \text{and} \quad i = j \star \delta_D(\alpha, q).$$

When the machine arrives at a cell g by the first time, it finds the state $c([g])$ on it. The proof follows as before. □

7. Conclusions and final comments

A class of subshifts is defined as the class of shifts which correspond to a TM. Several properties of this class are proved. A particular hierarchy is defined inside this class: for each $k \in \mathbb{N} \setminus \{0\}$, we consider the class of subshifts corresponding to a TM with tape $\mathbb{Z}^k$. It is proved that this hierarchy is infinite, and that it contains, for each $k \geq 2$, some subshifts that are not deterministic realtime. The classes of subshifts corresponding to TMs whose tape is a regular infinite tree of degree $2k$ was studied as well. It was proved that it contains some subshifts that are not in the previous hierarchy, for each k, and that it also contains non-deterministic realtime subshifts. It is left to characterize this class in a simple way, i.e., to find a simple necessary and sufficient condition for a subshift to correspond to some TM.

When comparing the power of a TM over a given infinite Cayley Graph and a TM over the line ($\mathbb{Z}$), we can assert that the first can always simulate the second, while the converse is not true in general. In fact, in order to simulate a TM over a Cayley Graph, it is necessary to solve, first, the word problem for the given graph, which may be undecidable. A TM over a Cayley Graph where the word problem is undecidable is more powerful than a TM over the line.

The class of machines whose subshift is sofic was studied. These machines present high restrictions on the movement ability of this class. These restrictions produce interesting dynamical behaviors when the machine starts over periodical initial configurations, as Theorems 1 and 2 establish. This result might be used to explain the eventually periodical behavior of Langton's Ant when it starts over an almost white configuration. But the subshift associated with Langton's Ant is not sofic, since the ant can make cycles of arbitrary length. Nevertheless, one can think that if the ant does not make long cycles during a long time, it will have a "sofic-like" behavior, and then it will fall into a periodic movement.

Acknowledgments

The first author wants to thank Petr Kůrka, Marie-Pierre Beal and Veronique Terrier for receiving her at their respective universities and for helpful conversations, as well as Maurice Nivat and Francesca Fiorenzi for fruitful discussions. First author's work was partially supported by CONICYT FONDECYT #1030706.

References

[1] M.-P. Béal, D. Perrin, Symbolic Dynamics and Finite Automata, in: G. Rosenberg, A. Salomaa (Eds.), Handbook of Formal Languages, vol. 2, Springer, 1997 (Chapter 10).
[2] V.D. Blondel, J. Cassaigne, C. Nichitiu, On the presence of periodic configurations in Turing machines and in counter machines, Theoret. Comput. Sci. 289 (2002) 573–590.
[3] M. Blum, W.J. Sakida, On the capability of finite automata in 2 and 3 dimensional space, in: 17th IEEE FOCS Conference, 1977, pp. 147–161.
[4] L.A. Bunimovich, S. Troubetzkoy, Topological dynamics of flipping Lorentz lattice gas models, J. Stat. Phys. 72 (1993) 297–307.
[5] M. Delorme, J. Mazoyer, Pebble Automata. Figures families recognition and universality, Research Report, No 99-32, 1999.
[6] A. Gajardo, A symbolic projection of Langton's Ant, in: Discrete Mathematics and Theoretical Computer Science Proceedings, vol. AB, 2003, pp. 57–68.
[7] A. Gajardo, E. Goles, Dynamics of a class of ants on a one-dimensional lattice, Theoret. Comput. Sci. 322 (2) (2004) 267–283.
[8] B. Kitchens, Symbolic Dynamics, Springer, 1998.
[9] P. Kůrka, On topological dynamics of Turing machines, Theoret. Comput. Sci. 174 (1–2) (1997) 203–216.
[10] P. Kůrka, A. Maass, Realtime subshifts, Theoret. Comput. Sci. 237 (2000) 307–325.
[11] P. Kůrka, Topological and Symbolic Dynamics, Société Mathématique de France, Paris, 2003.
[12] D. Lind, B. Marcus, An Introduction to Symbolic Dynamics and Coding, Cambridge University Press, 1995.